Task 1

library(data.table)

library(ggplot2)  
library(ggmosaic)

library(readr)

library(plyr)  
library(dplyr)

filePath = "C:/Users/jerem/Desktop/Quantium/"  
transactionData = fread(paste0(filePath,"QVI\_transaction\_data.csv"))  
customerData = fread(paste0(filePath,"QVI\_purchase\_behaviour.csv"))  
###Exploratory data analysis  
  
str(transactionData)

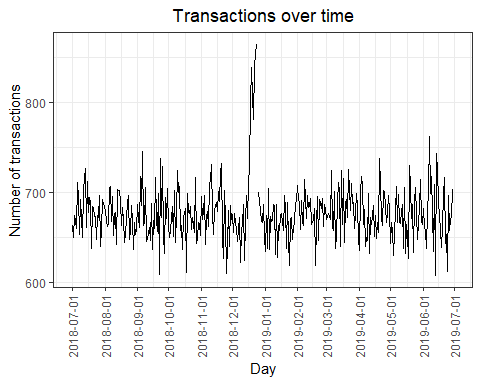
transactionData$DATE <- as.Date(transactionData$DATE, origin = "1899-12-30")  
  
# checking the type of product  
unique(transactionData$PROD\_NAME)

productWords <- data.table(unlist(strsplit(unique(transactionData[, PROD\_NAME]), "  
")))  
setnames(productWords, 'words')  
## removing digits and special characters  
  
productWords = apply(productWords, 2,function(productWords) gsub("[[:punct:]]", "",gsub("[[:digit:]]", "",gsub("\\g$","",tolower(productWords)))))  
  
##SORT PRODUCT WORDS BY OCCURENCE (NOT DONE YET)  
  
  
##Remove salsa products  
transactionData[, SALSA := grepl("salsa", tolower(PROD\_NAME))]  
transactionData = transactionData[SALSA == FALSE, ][, SALSA := NULL]  
  
  
##Summary stats  
summary(transactionData$PROD\_NBR)

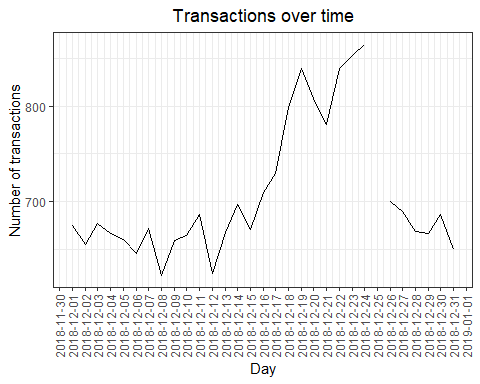
summary(transactionData$PROD\_QTY)

summary(transactionData$TOT\_SALES)

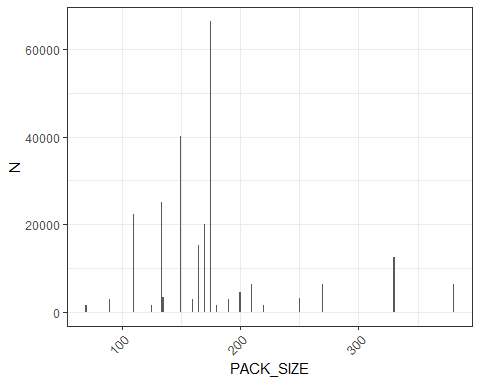
#finding and eliminating outlier  
outlier = subset(transactionData, transactionData$PROD\_QTY == 200)  
outlier\_ID = outlier$LYLTY\_CARD\_NBR[1]  
outlier\_transaction = subset(transactionData, transactionData$LYLTY\_CARD\_NBR == outlier\_ID)  
  
transactionData = transactionData[LYLTY\_CARD\_NBR != outlier\_ID]  
  
#Re-examining transaction Data  
unique\_Dates = as.data.frame(unique(transactionData$DATE))  
setnames(unique\_Dates, "Dates")  
  
# filling in missing day  
dates <- as.data.frame(seq(as.Date('2018-07-01'), as.Date('2019-06-30'), by = 'days'))  
setnames(dates, "DATE")  
transactions\_by\_dates = transactionData %>% group\_by(DATE) %>% count(DATE)  
Dates\_with\_missing = merge(transactions\_by\_dates, dates, by = "DATE", all.y = T)  
  
  
  
  
#### Setting plot themes to format graphs  
theme\_set(theme\_bw())  
theme\_update(plot.title = element\_text(hjust = 0.5))  
  
  
#### Plot transactions over time  
ggplot(Dates\_with\_missing, aes(x = DATE, y = n)) +  
 geom\_line() +  
 labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +  
 scale\_x\_date(breaks = "1 month") +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))



#plotting december  
december = Dates\_with\_missing %>% filter(DATE < as.Date("2019-01-01"))  
  
december = december %>% filter(DATE > as.Date("2018-11-30") )  
  
  
ggplot(december, aes(x = DATE, y = n)) +  
 geom\_line() +  
 labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +  
 scale\_x\_date(breaks = "1 day") +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))



# We can see that the increase in sales occurs in the lead-up to Christmas and that  
# there are zero sales on Christmas day itself. This is due to shops being closed on  
# Christmas day.  
# Now that we are satisfied that the data no longer has outliers, we can move on to  
# creating other features such as brand of chips or pack size from PROD\_NAME. We will  
# start with pack size.  
  
# pack size  
transactionData[, PACK\_SIZE := parse\_number(PROD\_NAME)]  
Pack\_size = transactionData[, .N, PACK\_SIZE][order(PACK\_SIZE)]  
#histogram  
  
plot\_pack\_size = ggplot(data= Pack\_size, aes(x=PACK\_SIZE, y= N))+  
geom\_bar(stat="identity")+  
theme(axis.text.x = element\_text(size = 10, angle = 45, hjust = 1))  
plot\_pack\_size

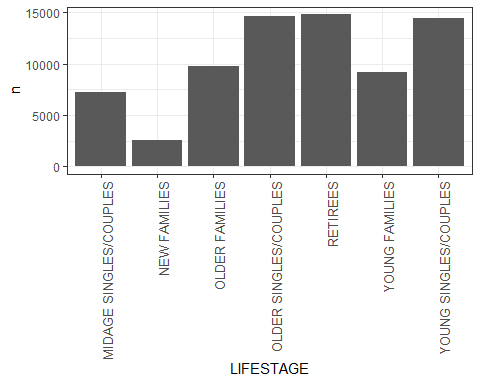


# brand\_name; combining the same brands.  
  
transactionData[, BRAND\_NAME := gsub("([A-Za-z]+).\*", "\\1",PROD\_NAME)]  
transactionData[BRAND\_NAME == "Red", BRAND\_NAME := "RRD"]  
unique(transactionData$BRAND\_NAME)

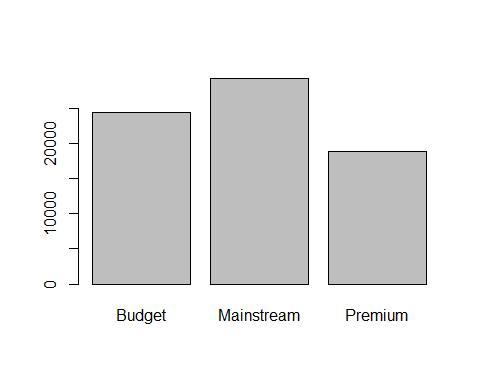
transactionData[BRAND\_NAME == "Dorito", BRAND\_NAME := "Doritos"]  
transactionData[BRAND\_NAME == "Snbts", BRAND\_NAME := "Sunbites"]  
transactionData[BRAND\_NAME == "Grain", BRAND\_NAME := "GrnWves"]  
transactionData[BRAND\_NAME == "WW", BRAND\_NAME := "Woolworths"]  
transactionData[BRAND\_NAME == "NCC", BRAND\_NAME := "Natural"]  
transactionData[BRAND\_NAME == "Infzns", BRAND\_NAME := "Infuzions"]  
transactionData[BRAND\_NAME == "Infzns", BRAND\_NAME := "Infuzions"]  
transactionData[BRAND\_NAME == "SMITH", BRAND\_NAME := "SMITHS"]  
# Examining customer Data  
  
str(customerData)

Customer\_by\_LifeStage = customerData %>% group\_by(LIFESTAGE) %>% count(LIFESTAGE)

plot\_lifestage = ggplot(data= Customer\_by\_LifeStage, aes(x=LIFESTAGE, y= n))+  
geom\_bar(stat="identity")+  
theme(axis.text.x = element\_text(size = 10, angle = 90, hjust = 1))  
plot\_lifestage

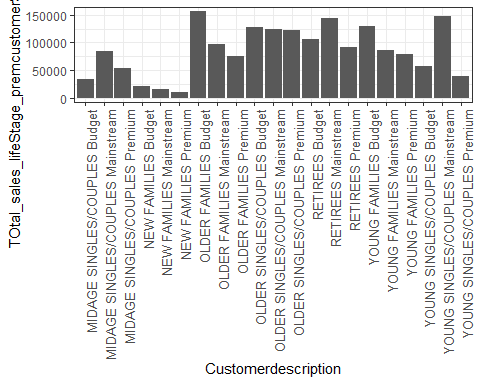


Customer\_by\_PREMstatus = customerData %>% group\_by(PREMIUM\_CUSTOMER) %>% count(PREMIUM\_CUSTOMER)  
barplot(height = Customer\_by\_PREMstatus$n, names.arg = Customer\_by\_PREMstatus$PREMIUM\_CUSTOMER)

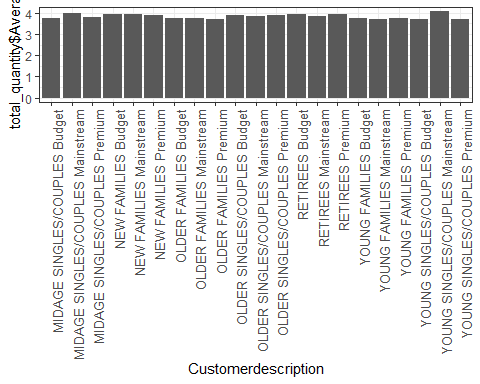


is.null(data)

# Merging data  
data <- merge(transactionData, customerData, all.x = TRUE)  
  
write.csv(data,"QVI\_data.csv")  
#PLotting data by premium customer and by life stage  
  
TOtal\_sales\_lifeStage\_premcustomer = aggregate(data$TOT\_SALES~data$LIFESTAGE + data$PREMIUM\_CUSTOMER, data, sum)  
TOtal\_sales\_lifeStage\_premcustomer$Customerdescription = paste(TOtal\_sales\_lifeStage\_premcustomer$`data$LIFESTAGE`, TOtal\_sales\_lifeStage\_premcustomer$`data$PREMIUM\_CUSTOMER`)  
plot = ggplot(data= TOtal\_sales\_lifeStage\_premcustomer, aes(x=Customerdescription, y= TOtal\_sales\_lifeStage\_premcustomer$`data$TOT\_SALES`))+  
geom\_bar(stat="identity")+  
theme(axis.text.x = element\_text(size = 10, angle = 90, hjust = 1))  
plot



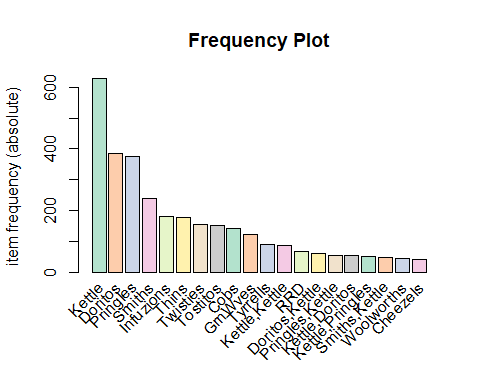
total\_quantity = aggregate(data$PROD\_QTY~data$LIFESTAGE + data$PREMIUM\_CUSTOMER, data, sum)  
total\_quantity = as.data.frame(aggregate(data$PROD\_QTY~data$LIFESTAGE + data$PREMIUM\_CUSTOMER, data, sum))  
total\_quantity$Customerdescription = paste(total\_quantity$`data$LIFESTAGE`, total\_quantity$`data$PREMIUM\_CUSTOMER`)  
total\_quantity = subset(total\_quantity, select = c("Customerdescription", "data$PROD\_QTY"))  
total\_quantity = merge(total\_quantity, TOtal\_sales\_lifeStage\_premcustomer, by = "Customerdescription")  
total\_quantity$Average = total\_quantity$`data$TOT\_SALES`/total\_quantity$`data$PROD\_QTY`  
  
  
plot\_average\_sales = ggplot(data= total\_quantity, aes(x=Customerdescription, y= total\_quantity$Average))+  
geom\_bar(stat="identity")+  
theme(axis.text.x = element\_text(size = 10, angle = 90, hjust = 1))  
plot\_average\_sales



#### t test  
  
t.test(data[PREMIUM\_CUSTOMER == "Mainstream" &   
 LIFESTAGE %in% c("YOUNG SINGLES/COUPLES",  
 "MIDAGE SINGLES/COUPLES"), data$TOT\_SALES/data$PROD\_QTY],data[ PREMIUM\_CUSTOMER != "Mainstream" & LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES"), data$TOT\_SALES/data$PROD\_QTY], alternative = "greater")

## Deep dive into specific customer segments for insights  
library(arules)

library(RColorBrewer)  
main\_young = data[PREMIUM\_CUSTOMER == "Mainstream" & LIFESTAGE == "YOUNG SINGLES/COUPLES"]  
  
main\_young\_PROD\_NAME = ddply(main\_young,c("LYLTY\_CARD\_NBR"),  
 function(dataframe)paste(dataframe$PROD\_NAME,  
 collapse = ","))  
  
main\_young\_BRAND = ddply(main\_young,c("LYLTY\_CARD\_NBR"),  
 function(dataframe)paste(dataframe$BRAND\_NAME,  
 collapse = ","))  
write.csv(main\_young\_BRAND,"main\_young\_brand.csv")  
main\_young\_transaction <- read.transactions("C:/Users/jerem/Documents/R/Quantium/Quantium/main\_young\_brand.csv", format = 'basket', sep=',')  
itemFrequencyPlot(main\_young\_transaction,topN = 20,type = "absolute",col=brewer.pal(8,'Pastel2'), main="Frequency Plot", angle = 90)



main\_young\_SIZE = ddply(main\_young,c("LYLTY\_CARD\_NBR"),  
 function(dataframe)paste(dataframe$PACK\_SIZE,  
 collapse = ","))  
  
write.csv(main\_young\_SIZE,"main\_young.csv")  
main\_young\_transaction <- read.transactions("C:/Users/jerem/Documents/R/Quantium/Quantium/main\_young.csv", format = 'basket', sep=',')  
itemFrequencyPlot(main\_young\_transaction,topN = 20,type = "absolute",col=brewer.pal(8,'Pastel2'), main="Frequency Plot")

